

# Automatic design of robot programs

Andrea Roli

Dipartimento di Informatica–Scienza e Ingegneria (DISI)

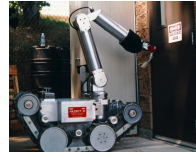
Campus of Cesena

*Alma Mater Studiorum* Università di Bologna

andrea.roli@unibo.it

Cesena, 31 May 2013

# Robots everywhere: Issues



## Cognitive systems and robotics EU research programme

“Hard scientific problems still have to be worked on in order to make robotic devices fit for rendering meaningful services to people”

(ERCIM News, January 2011)

## By 2020, robots should:

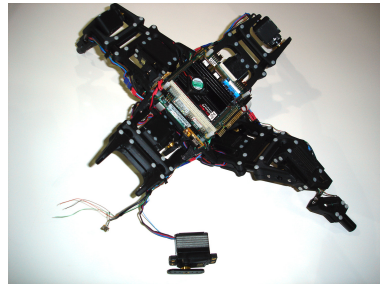
Be programmable by learning



(Brenna D. Argall, Northwestern University, with ICUB robot)

## By 2020, robots should:

Be able to predict failure states  
in their operating environment  
and their own bodies



(Starfish, by Hod Lipson,  
Cornell University)

# By 2020, robots should:

Plan complex tasks



# By 2020, robots should:

Be able to deal with novelty,  
uncertainty and change.



From Pfeifer & Scheier, 'Understanding intelligence',  
The MIT Press, 1999

## By 2020, robots should:

Be able to deal with novelty,  
uncertainty and change.



From Pfeifer & Scheier, 'Understanding intelligence',  
The MIT Press, 1999

In other terms, they should be both **robust** and **adaptive**

# Outline

- 1 Automatic design
- 2 Evolutionary robotics
- 3 FSA optimisation
- 4 Boolean network robotics
- 5 Case studies
  - Phototaxis–antiphototaxis
  - Obstacle avoidance
  - Sequence learning task
  - Swarm of BN-robots



# Outline

- 1 Automatic design
- 2 Evolutionary robotics
- 3 FSA optimisation
- 4 Boolean network robotics
- 5 Case studies
  - Phototaxis–antiphototaxis
  - Obstacle avoidance
  - Sequence learning task
  - Swarm of BN-robots

# Automatic design: motivations

- Available technique might not be suitable for some kinds of goals (e.g., tasks in swarm robotics)
- It is not possible to formally and precisely define the robot system task
- The task can vary in time
- Dynamic and uncertain environment
- The designer wants to explore new solutions
- The scientist wants to investigate the emergence of some kinds of behaviour

# Main approaches for automatic design

- The robot program is represented with a formal language (e.g., FSA)
- The design problem is encoded into a learning problem
- Usually, in the training phase an optimisation algorithm is used (e.g., evolutionary techniques)
- The program is designed in simulation and then tested on a real setting

# Issues

- 1 Training and test set
- 2 Merit factor (i.e., objective function or fitness function)
- 3 Simulation

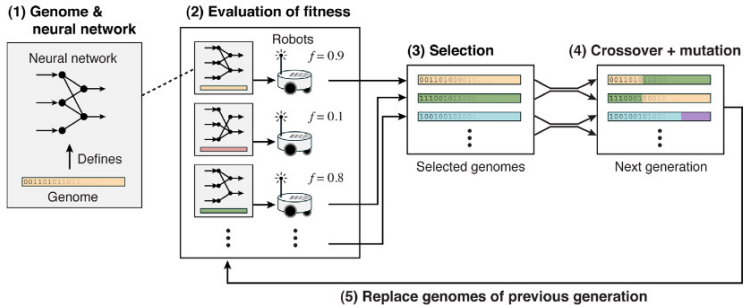
# Current approaches

- 1 Evolutionary robotics
- 2 *Finite state automaton* optimisation (with evolutionary algorithms or other kinds of search)
- 3 Boolean network robotics
- 4 Reinforcement learning

# Outline

- 1 Automatic design
- 2 Evolutionary robotics**
- 3 FSA optimisation
- 4 Boolean network robotics
- 5 Case studies
  - Phototaxis–antiphototaxis
  - Obstacle avoidance
  - Sequence learning task
  - Swarm of BN-robots

# Evolutionary robotics



(From Floreano and Keller, *Evolution of Adaptive Behaviour in Robots by Means of Darwinian Selection*,

PLOS Biology, Jan. 10, Vol. 8, Issue 1)

# Evolutionary robotics

See also

Relevant works by:

- Josh Bongard
- Hod Lipson
- Andrew L. Nelson
- Stefano Nolfi
- Vito Trianni



# Outline

- 1 Automatic design
- 2 Evolutionary robotics
- 3 FSA optimisation**
- 4 Boolean network robotics
- 5 Case studies
  - Phototaxis–antiphototaxis
  - Obstacle avoidance
  - Sequence learning task
  - Swarm of BN-robots

# FSA optimisation

Main idea:

- Robot program is a FSA
- The structure of the FSA is optimised so as to maximise a merit factor
- Issues: definition of merit factor, definition of primitives of FSA

## Example in Swarm robotics

Work by Gianpiero Francesca and Mauro Birattari (IRIDIA-ULB)

- Primitive pre-coded behaviours (e.g., random walk)
- Parametrised conditions (for transitions between states)
- Search in the space of FSA by means of *Iterated F-race*

Link:

[http://iridia.ulb.ac.be/~gfrancesca/Gianpiero\\_Francesca/Research.html](http://iridia.ulb.ac.be/~gfrancesca/Gianpiero_Francesca/Research.html)

# Outline

- 1 Automatic design
- 2 Evolutionary robotics
- 3 FSA optimisation
- 4 Boolean network robotics**
- 5 Case studies
  - Phototaxis–antiphototaxis
  - Obstacle avoidance
  - Sequence learning task
  - Swarm of BN-robots

# The subject

Dynamical system theory and complexity science are rich sources for:

- analysing artificial agents and robots
- design principles and guidelines

# The subject

Dynamical system theory and complexity science are rich sources for:

- analysing artificial agents and robots
- design principles and guidelines

## Boolean network robotics

Boolean network robotics concerns the use of Boolean networks, and other models from complex systems science, as robot programs.



# Boolean networks

- Introduced by Stuart Kauffman in 1969 as a genetic regulatory network (GRN) model
- Discrete-time / discrete-state dynamical system
- Non trivial (*complex*) dynamics



# Structure

- System composed of  $N$  Boolean variables  $x_1, x_2, \dots, x_N$
- Each variable  $x_i$  is associated to a Boolean function  
 $f_i(x_{i_1}, x_{i_2}, \dots, x_{i_{K_i}})$

# Dynamics

- System state at time  $t$ :  $s(t) = (x_1(t), \dots, x_N(t))$
- $x_i(t+1) = f_i(x_{i_1}, x_{i_2}, \dots, x_{i_{K_i}})$
- Dynamics controls node update
- Several kinds of dynamics

# Dynamics

- System state at time  $t$ :  $s(t) = (x_1(t), \dots, x_N(t))$
- $x_i(t+1) = f_i(x_{i_1}, x_{i_2}, \dots, x_{i_{K_i}})$
- Dynamics controls node update
- Several kinds of dynamics

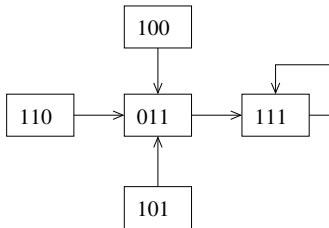
## Synchronous dynamics and deterministic update rules:

- One successor per state
- Cardinality of state space  $2^N$

# Dynamics

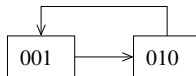
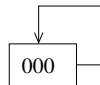
Trajectory composed of:

- Transient
- Attractor



Attractors:

- Fixed points
- Cycles



# Relevance of BNs

- Minimal complex system
- Several important phenomena in genetics can be reproduced  
(e.g., random BN models for reproducing KO gene expression avalanches and cell differentiation)
- Tight connections with the satisfiability problem

# Motivation

- Information processing  $\leftrightarrow$  evolution in time of a dynamical system
- GRNs are very compact, yet they produce complex behaviours
- Cells able to balance robustness and adaptivity

# Motivation

- Information processing  $\leftrightarrow$  evolution in time of a dynamical system
- GRNs are very compact, yet they produce complex behaviours
- Cells able to balance robustness and adaptivity



GRN models could be engineered and used as robot programs  
(mainly when robots are asked to be robust, adaptive and evolvable)

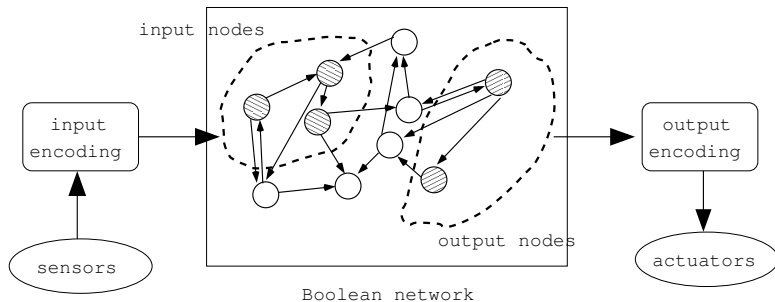
# Design Boolean network robots

Main issues:

- 1 Definition of the mapping between sensors/actuators and network's inputs/outputs
- 2 Design the BN that serves as robot program



# BN-robot coupling



# BN design

## The approach

Transforming design into a constrained optimisation problem.

We aim at designing a BN such that:

- its dynamics satisfies given requirements
- the robot's performance is maximised

# BN design

## The approach

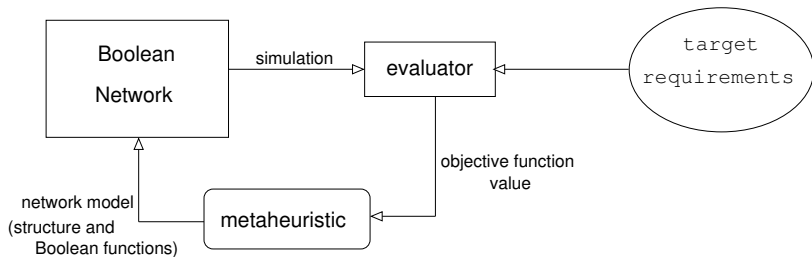
Transforming design into a constrained optimisation problem.

We aim at designing a BN such that:

- its dynamics satisfies given requirements
- the robot's performance is maximised

→ Metaheuristic techniques (a.k.a. stochastic local search techniques)

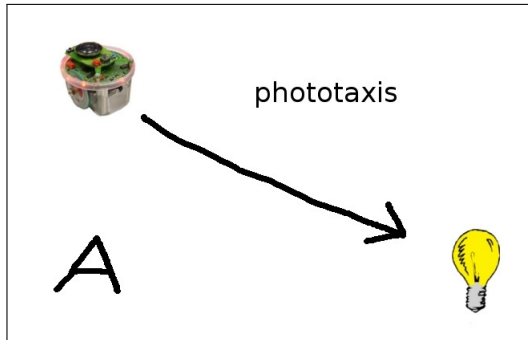
# BN design by metaheuristics



# Outline

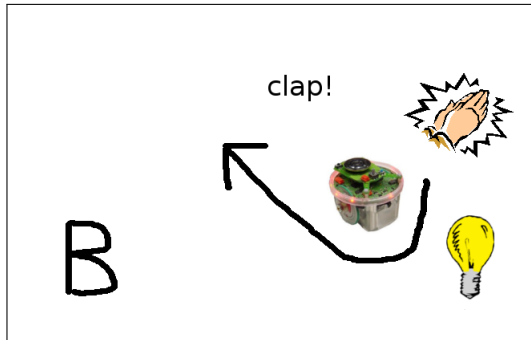
- 1 Automatic design
- 2 Evolutionary robotics
- 3 FSA optimisation
- 4 Boolean network robotics
- 5 Case studies**
  - Phototaxis–antiphototaxis
  - Obstacle avoidance
  - Sequence learning task
  - Swarm of BN-robots

# The task



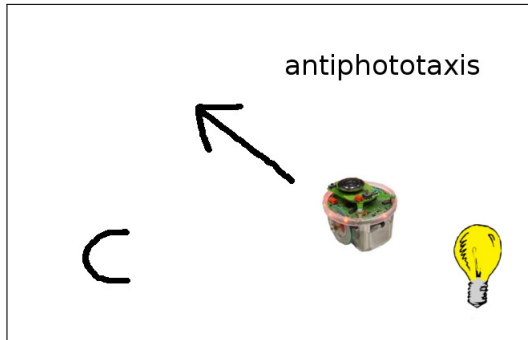
The robot seeks the light and goes towards it

# The task



After perceiving a sharp sound it turns. . .

# The task



...and goes away from the light



# The task

- Typical task in evolutionary robotics
- The robot has to learn two different behaviours...
- ...and to keep memory of a specific event to discriminate between the two behaviours
- We also extended the task to 2 and 3 claps

# The robot

## e-puck robot (EPFL)

- two wheels
- light sensors
- sound sensor
- plus other sensor and actuators



# BN-robot setup

## ■ BN with 20 nodes

Input and outputs:

- $x_1 \leftrightarrow$  sound
- $x_2, x_3, x_4, x_5 \leftrightarrow$  light sensors (N,S,E,W)
- $x_6, x_7 \leftrightarrow$  right and left wheels  
(postprocessing with shifting window average)

# Training and testing

- Training: simulation
- Testing: simulation and real robot
- Noise on sensors added also in training

**Incremental training:** first BN-robot trained to perform only phototaxis, then trained to accomplish the whole task.

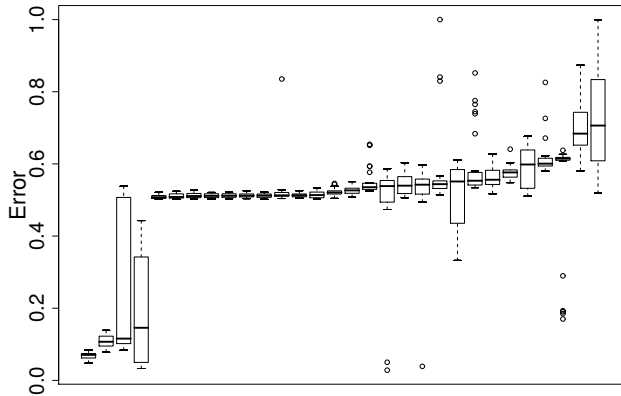
# Metaheuristic algorithm

- Adaptive walk (aka stochastic descent)
- Move: one flip in one function's truth table (randomly chosen)
- Objective function evaluates the robot's error in accomplishing the task
- Initial BNs randomly generated with  $K=3$ . Random topology.

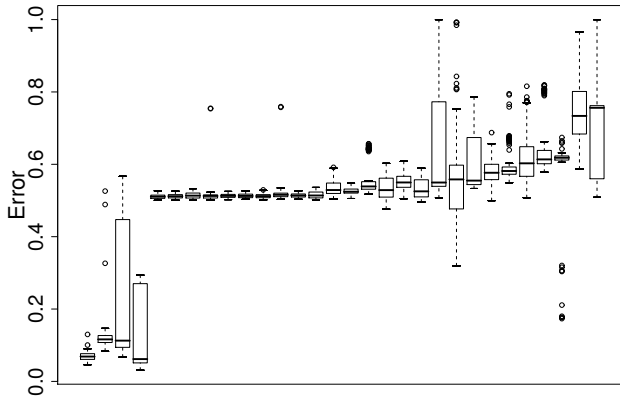
# Experimental setting

- Environment: square arena (1m x 1m) with a light source in one corner
- Evaluation: average over 30 different scenarios
  - Robot placed in a random position close to the opposite corner
  - Robot initial rotation is random
  - Sound emitted in a random instant
- 30 independent experiments

# Results: training



# Results: testing





# Video



## BN-robot dynamics

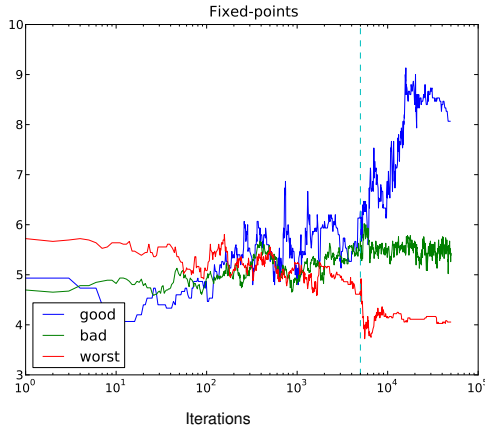
- Analysis of trajectories in the BN state space during the design process
- BNs make it possible to directly analyse system's *symbolic dynamics*
- Same results in simulation and real testing

# BN-robots' symbolic dynamics

Features of interest:

- Number of fixed points → micro-behaviours
- *Complexity* of the trajectory → computational capabilities

# Number of fixed points

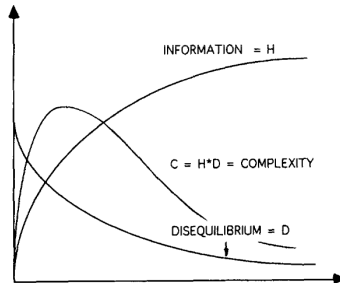


**Emergence of  
*micro-behaviours***  
(segmentation of  
behaviour)

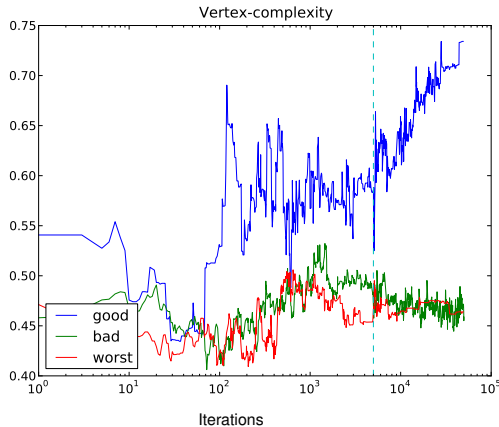
# Complexity

## Statistical complexity of BN states

$$\text{Complexity} = \text{Entropy} \times \text{Disequilibrium}$$



# Complexity



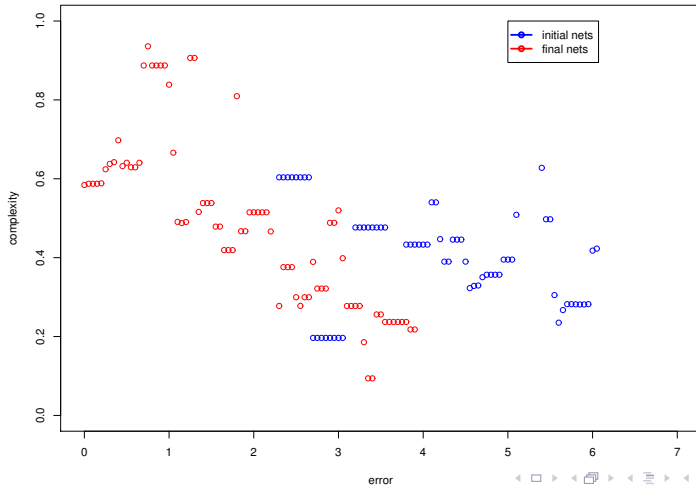
**Computational  
capability**

# The task

The robot walks along a corridor and must reach the end by avoiding walls and obstacles along the corridor

- *e-puck* robot with 4 proximity sensors (NE,SE,SW,NW) and two wheels (differential)
- Task learned easily by a BN-robot

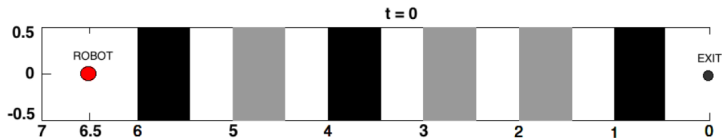
# Complexity of controllers





# The task

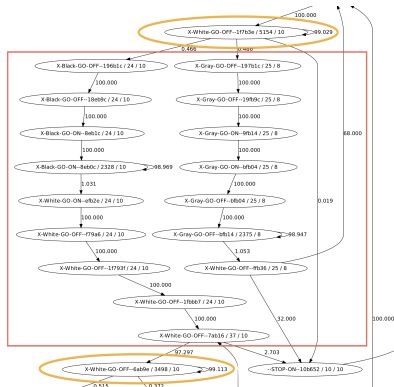
The robot walks along a corridor and must switch-on the LEDs when the colour it perceives is in the right order in the given sequence.





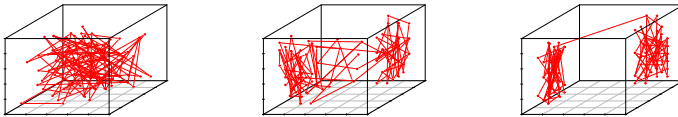
# Analysis of trajectories in the BN state space

Trajectories are collected in the BN state space while robots acts and their characteristics analysed



# Results

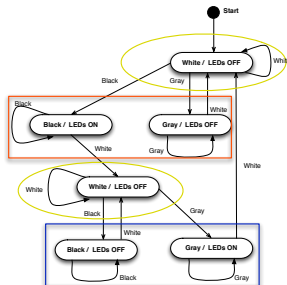
Trajectories cluster in the phase space depending on the symbol to be remembered



Metaheuristic algorithm iterations

## Results

- A **finite state automaton** can be derived representing the abstract behaviour of the network
- FSA states are implicitly represented by areas in the BN phase space

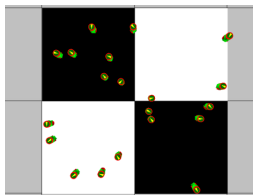


# The task

- Distributed pattern recognition (as a simplified case of distributed environment processing)
- The BN-robots should reach a consensus on the class of environment they are located by exchanging only local information
- All the robots have the same BN-controller

# Results

- Tested on two different settings, each with two ground patterns
- Robots placed randomly (also rotated randomly)
- Importance of exchanging both the ground sensor information and the ‘individual answer’



## Ongoing work

- Towards more complex tasks
- General method for analysing trajectories
- Exploit the complexity of GRN dynamics (e.g., cell differentiation mechanisms, epigenetic control)
- Use models such as Probabilistic BNs, multi-valued BNs and Glass networks

BN Robotics website

<http://iridia.ulb.ac.be/bn-robotics/>